IN THE CLAIMS

Please amend claims 43, 50, 52, 54, 56, 119, 122, 123, and 126 as indicated below.

1-42. (Cancelled)

43. (Currently Amended) A system comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator marks variables associated with the stack-based instructions in the register file as modified when the variables are updated as a result of the processing of the stack-based instructions to enable selective writing of the variables marked as modified to a memory~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

44-49. (Cancelled)

50. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator generates a new ~~Java~~virtual machine program counter due to a "GOTO" or "GOTO_W" byte code by sign extending the immediate branch offset following the "GOTO" or "GOTO_W" byte code and adds it to the ~~Java~~virtual machine program counter of the current byte code instruction~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

51. (Cancelled)

52. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator generates a new ~~Java~~virtual machine program counter due to a "JSR" or "JSR_W" byte code by sign extending the immediate branch offset following the "JSR" or "JSR_W" byte code and adding it to the ~~Java~~virtual machine PC of the current byte code instruction, computes the return ~~Java~~virtual machine program counter and pushes the return ~~Java~~virtual machine program counter onto an operand stack~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

53. (Cancelled)

54. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core; wherein the hardware accelerator performs sign extension for the ~~Java~~virtual machine SiPush and BiPush byte codes and appends the sign extended data to the immediate field of a register-based instruction being composed based the stack-based instructions~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

55. (Cancelled)

56. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator performs sign extension for the ~~Java~~virtual machine SiPush and BiPush byte codes and makes the sign extended data available to be read by the CPU core~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

57-108. (Cancelled)

109. (Original) The system of claim 43, wherein the hardware accelerator and the CPU core are within a CPU.

110. (Original) The system of claim 43, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

111. (Original) The system of claim 50, wherein the hardware accelerator and the CPU core are within a CPU.

112. (Original) The system of claim 50, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

113. (Original) The system of claim 52, wherein the hardware accelerator and the CPU core are within a CPU.

114. (Original) The system of claim 52, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

115. (Original) The system of claim 54, wherein the hardware accelerator and the CPU core are within a CPU.

116. (Original) The system of claim 54, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

117. (Original) The system of claim 56, wherein the hardware accelerator and the CPU core are within a CPU.

118. (Original) The system of claim 56, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

119. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator maintains an operand stack for the stack-based instructions in the register file such that the operand stack in the register file define a ring buffer in conjunction with an overflow/underflow mechanism for moving operands in the operand stack between the register file and memory, and loads variables required for processing the stack-based instructions into the register file~~; and a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

120. (Original) The system of claim 119, wherein the hardware accelerator and the CPU core are within a CPU.

121. (Original) The system of claim 119, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

122. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core; wherein the hardware accelerator maintains operands and variables required for processing the stack-based instructions in the register file~~; and a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

123. (Currently Amended) The system of claim 122, wherein the stack-based instructions comprise ~~Java~~virtual machine instructions.

124. (Original) The system of claim 122, wherein the hardware accelerator and the CPU core are within a CPU.

125. (Original) The system of claim 122, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

126. (Currently Amended) A system, comprising:

a central processing unit (CPU);

a register file associated with the CPU core; and

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator:

maintains an operand stack for the stack-based instructions in the register file such that the operand stack in the register file define a ring buffer in conjunction with an overflow/underflow mechanism for moving operands in the operand stack between the register file and memory, and loads variables required for processing the stack-based instructions into the register file,

generates a new ~~Java~~virtual machine program counter due to a "GOTO" or "GOTO_W" byte code by sign extending the immediate branch offset following the "GOTO" or "GOTO_W" byte code and adds it to the ~~Java~~virtual machine program counter of the current byte code instruction,

generates a new ~~Java~~virtual machine program counter due to a "JSR" or "JSR_W" byte code by sign extending the immediate branch offset following the "JSR" or "JSR_W" byte code and adding it to the ~~Java~~virtual machine PC of the current byte code instruction, computes the return ~~Java~~virtual machine program counter and pushes the return ~~Java~~virtual machine program counter onto the operand stack,

performs a sign extension for the ~~Java~~virtual machine SiPush and BiPush byte codes and appends the sign extended data to the immediate field of a register-based instruction being composed based the stack-based instructions,

performs sign extension for the ~~Java~~virtual machine SiPush and BiPush byte codes and makes the sign extended data available to be read by the CPU core, and

marks variables associated with the stack-based instructions in the register file as modified when the variables are updated as a result of the processing of the stack-based instructions to enable selective writing of the variables marked as modified to a memory~~; and~~

~~a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core~~.

127. (Original) The system of claim 126, wherein the hardware accelerator and the CPU core are within a CPU.

128. (Original) The system of claim 126, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.